**Mutabilidad en el contexto de Python**

En Python una variable inmutable es aquella variable cuyo valor no puede ser modificado después de ser creada. Por ejemplo: int, float, tuple y string.

Una variable inmutable es lo contrario, es aquella variable cuyo valor puede ser modificado después de ser creada como: list, dict, set y array.

Esto hace que cada tipo de dato pueda tener usos en ciertos casos dónde es más eficiente que otro.

●  Los tipos inmutables son generalmente más rápidos de acceder.

●  Los tipos mutables son perfectos cuando quieres cambiar su contenido repetidas veces.

●  Los tipos inmutables son caros de cambiar, ya que lo que se hace en realidad es

**Ejemplo de inmutabilidad**

>> name = “laura”

>> name[1] = “o”

ERROR “**‘str’ object does not support item assignment**”

>>name

“laura”

En este caso se le asignó un string “laura” a la variable name luego, se intentó cambiarle la segunda letra del nombre y arrojó un error, el cual menciona que (el objeto “str” no admite la asignacion de elementos), esto quiere decir que el string es inmutable por ende no permite modificarlo una vez creado.

**Ejemplo de mutabilidad:**

>> name [“L”; “a” “u”; “r”; “a”]

>>name

[“L”; “a” “u”; “r”; “a”]

>>name[1] = “o”

>>name

[[“L”; “o” “u”; “r”; “a”]

En este ejemplo se le asigno los caracteres “L” “a” u” ”r” ”a” al nombre de la lista

Y se intentó cambiar la segunda letra, el cual resultó. Esto es debido a que las listas son mutable, es decir, pudo ser modificado despues de ser creado.